

1 SINGER CASHMAN LLP
Benjamin L. Singer (Bar. No. 264295)
2 bsinger@singercashman.com
Evan N. Budaj (Bar No. 271213)
3 ebudaj@singercashman.com
4 601 Montgomery Street, Suite 1950
San Francisco, California 94111
5 Telephone: (415) 500-6080
6 Facsimile: (415) 500-6080
Attorneys for Plaintiff Software Research, Inc.

7 UNITED STATES DISTRICT COURT
8
9 NORTHERN DISTRICT OF CALIFORNIA
10 SAN FRANCISCO DIVISION

11 SOFTWARE RESEARCH, INC.,
12 Plaintiff,
13 v.
14 PROGRESS SOFTWARE CORPORATION,
15 TELERIK INC., and DOES 1 through 10,
16 Defendants.

CASE NO. 3:20-CV-1843

**COMPLAINT FOR PATENT
INFRINGEMENT**

JURY TRIAL DEMANDED

1 Plaintiff Software Research, Inc. (“SRI”), for its Complaint against Progress Software
2 Corporation (“Progress Software”), Telerik Inc. (“Telerik”), as well as Does 1 through 10
3 (collectively, along with Progress Software and Telerik, “Defendants”), upon information and belief,
4 state and allege as follows:

5 **NATURE OF THE ACTION**

6 1. This is a civil action for patent infringement arising under the patent laws of the
7 United States, Title 35 of the United States Code.

8 2. As set forth in more detail below, Defendants have been infringing United States
9 Patent Nos. 7,757,175 (the “’175 Patent”); 8,327,271 (the “’271 Patent”); 8,392,890 (the “’890
10 Patent”); 8,495,585 (the “’585 Patent”); 8,650,493 (the “’493 Patent”), 8,984,491 (the “’491 Patent”)
11 and 10,489,286 (the “’286 Patent”) (collectively, the “Patents-in-Suit”), and continue to do so
12 through the present date.

13 **THE PARTIES**

14 3. SRI is a corporation organized and existing under the laws of the State of California
15 with its principal place of business in this District.

16 4. Upon information and belief, Progress Software is a Delaware corporation with its
17 principal place of business at 14 Oak Park Drive, Bedford, Massachusetts 01730 and an office at 203
18 Redwood Shores Parkway, Redwood City, California 94065.

19 5. Upon information and belief, Telerik is a wholly owned subsidiary of Progress
20 Software and a Delaware corporation with its principal place of business at 201 Jones Road,
21 Waltham, Massachusetts 02451.

22 6. Upon information and belief, Defendants Does 1 through 10 are directors, officers,
23 employees, representatives, and/or agents of Progress Software and/or Telerik who participated
24 and/or are currently participating in the use, development, sale, offer for sale, import, offer for
25 import, and/or other commercialization of software offerings that infringe one or more of the Patents-
26 in-Suit. The true identities of Defendants Does 1 through 10 are presently unknown to SRI; SRI will
27 amend its complaint to state such names when they become known to SRI through discovery and/or
28 continued investigation.

7. Unless specifically stated otherwise, the acts complained of herein were committed by, on behalf of, and/or for the benefit of Progress Software and Telerik.

JURISDICTION AND VENUE

8. This Court has subject matter jurisdiction pursuant to 28 U.S.C. §§ 1331 and 1338(a).

9. This Court has personal jurisdiction over Defendants because (a) they reside in this State and this District, (b) they transact business in this State and this District, (c) they have committed the acts of patent infringement complained of herein, including but not limited to offering for sale or selling infringing products embodying SRI's patented invention, in this State and this District, and/or (d) they have directed their acts of infringement and the other unlawful acts complained of herein at this State and this District.

10. This Court has personal jurisdiction over Defendants for the additional reason that they have engaged in systematic and continuous contacts with this State and this District by, *inter alia*, regularly conducting and soliciting business in this State and this District, and deriving substantial revenue from products and/or services provided to persons in this State and this District.

11. Venue is proper in this District under 28 U.S.C. § 1391(b) because a substantial part of the acts complained of herein occurred in this District, Progress Software and Telerik transact business in this District, Progress Software and Telerik reside in this District for purposes of venue, and/or the property that is the subject of this action is situated in this District.

12. With respect to Progress Software and Telerik, venue is proper in this District under 28 U.S.C. §§ 1391(c)-(d) and 1400(b) because (i) Progress Software and Telerik reside in this District for purposes of venue; (ii) Progress Software and Telerik have committed acts of infringement in this District; and (iii) Progress Software has a regular and established place of business in this District.

BACKGROUND

13. Defendants develop web application monitoring and scripting tool software products known as, upon information and belief, the Test Studio (hereinafter, "Test Studio"). *See* <https://www.telerik.com/teststudio>.

14. Defendants offer for sale and sell the Test Studio to the public.

15. Defendants use the Test Studio, including at least in order to test the Test Studio as part of their development efforts.

16. The Test Studio is a software testing tool used to automate testing of software applications, including web-based applications.

17. Defendants claim “With Progress Telerik Test Studio you could save up to 50% in testing time.” See <https://www.telerik.com/teststudio>.

18. The Test Studio includes functionality for creating, storing, and executing test scripts. See <https://docs.telerik.com/teststudio/introduction/for-qa-testers> (“Test Studio redefines the way automated testing is done. Build your tests in minutes instead of hours! No need to waste time on defining coordinates in dynamic pages anymore. Record your test just once and played on multiple browsers without re-recording.”).

COUNT I – INFRINGEMENT OF THE '175 PATENT

19. SRI re-alleges and incorporates the allegations of the preceding paragraphs of this Complaint as if fully set forth herein.

20. SRI is the assignee and owner of all right, title, and interest in and to the '175 Patent, which was issued on July 13, 2010. A true and correct copy of the '175 Patent is attached hereto as Exhibit A.

21. The '175 Patent addresses an invention for testing websites. This disclosed innovation tests many facets of the website’s experience and operation, including by providing novel approaches to creating, storing, and executing test scripts using website elements as opposed to the previously disclosed use of recording test scripts based upon user actions only.

22. SRI has the exclusive right to make, use, sell, and offer to sell any product embodying the '175 Patent throughout the United States, and to import any product embodying the '175 Patent into the United States.

23. SRI has commercially exploited the '175 Patent by making, marketing, selling, and using products covered by the '175 Patent, including its popular eValid™ software products. SRI continues to commercially exploit the '175 Patent through the present, at least by continuing to provide maintenance and support to users of its popular eValid™ software products.

24. Defendants have had knowledge of the '175 Patent, SRI, and SRI's products embodying the inventions claimed in the Patents-in-Suit since at least as early as the filing of this Complaint.

25. At all relevant times, SRI provided public notice of the '175 Patent at least by properly marking its products and its website pursuant to 35 U.S.C. § 287(a).

26. Defendants have been, and are currently, directly infringing at least claim 11 of the '175 Patent in violation of 35 U.S.C. § 271(a), literally or under the doctrine of equivalents, by making, using, selling, offering for sale, and/or importing into the United States certain website testing software, including without limitation Defendants' web application monitoring and scripting tool software products titled, upon information and belief, the Test Studio and/or other related software products and services offered by Defendants (Defendants' "Infringing Products"), which, as set forth in documentation available on Defendants' websites, comprise the non-transitory computer readable media disclosed in the '175 Patent—both as maintained in Defendants' files and as made accessible to its users to whom Defendants offer and sell the Infringing Products—including at least computer program code stored therein for providing a test-enabled web browser for operation on a computing device to test a website hosted by a remote server, the website having at least one webpage (for example, "Progress expertise in web technologies and leadership in UI component development is leveraged through our automated software testing tool: Progress Test Studio, to assist users in automating even the most complex web testing scenarios.")

(<https://www.telerik.com/teststudio/html-testing>); the Test Studio as used with a web browser is a test-enabled web browser that can be used for "Web applications testing"

(<https://www.telerik.com/teststudio/html-testing>; see also

<https://docs.telerik.com/teststudio/welcome>) "Test Studio functional testing is a comprehensive yet cost-effective automated testing suite."); the website, necessarily including at least one webpage,

necessarily resides on a remote server and Defendants utilize any number of browsers as its "test-

enabled web browser" ([https://docs.telerik.com/teststudio/general-information/test-](https://docs.telerik.com/teststudio/general-information/test-recording/overview)

[recording/overview](https://docs.telerik.com/teststudio/general-information/test-recording/overview); <https://www.telerik.com/teststudio/html-testing>); web browsing components (for

example, the Test Studio allows a user to browse the web via common web browsing activities,

1 including navigating to a website and firing events such as clicking on a button)
 2 (<https://docs.telerik.com/teststudio/features/recorder/overview>;
 3 <https://docs.telerik.com/teststudio/general-information/test-recording/overview>;
 4 <https://docs.telerik.com/teststudio/getting-started/first-project#test-recording>); a page evaluation
 5 component that operates to read, extract, and analyze and confirm the contents of page components,
 6 including Document Object Model (DOM) elements with their associated at least one index and their
 7 values (for example, the Test Studio allows for the creation of test scripts to test websites by
 8 recording a user's interactions with the webpage in question and allowing the user to play back those
 9 test scripts) (<https://docs.telerik.com/teststudio/getting-started/first-project#test-recording>;
 10 <https://docs.telerik.com/teststudio/general-information/test-execution/quick-execution>); the Test
 11 Studio interrogates the DOM to identify and extract relevant information regarding at least the page
 12 elements germane to the script, including each such element's index and value, and stores those
 13 details in the test script ([https://docs.telerik.com/teststudio/general-information/test-results/analyze-
 14 test-list-results](https://docs.telerik.com/teststudio/general-information/test-results/analyze-test-list-results)); these page elements are located based on their DOM indexes (the Test Studio must
 15 necessarily use the DOM access methods included in Dynamic Linked Libraries associated with a
 16 browser code library) (<https://docs.telerik.com/teststudio/features/recorder/dom-explorer>;
 17 [https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-
 18 wtc/element-identification-wtc/finding-page-elements](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements); [https://docs.telerik.com/teststudio/testing-
 19 framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-
 20 elements](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements); <https://docs.telerik.com/teststudio/general-information/configure-your-browser/edge>); a test
 21 data component that operates to store facts about the at least one webpage (for example, the Test
 22 Studio uses explicit wait commands that “wait for the comparison to be true before proceeding” and
 23 in order to perform such a validation, the Test Studio must necessarily store facts about the webpage
 24 being rendered, i.e., the expected condition to be checked for during validation)
 25 (<https://docs.telerik.com/teststudio/features/recorder/verifications/wait>;
 26 <https://www.telerik.com/blogs/explicit-waits-with-ajax-combobox-example>); and a graphical user
 27 interface to provide user access to at least said web browsing components and at least one of said
 28 page evaluation components and said test data component (the Test Studio has a graphical user

1 interface to provide user access to test details including videos and logs of test results
 2 (<https://docs.telerik.com/teststudio/general-information/test-results/analyze-test-list-results>)), as
 3 disclosed in the '175 Patent.

4 27. Defendants will, on information and belief, continue to directly infringe the '175
 5 Patent unless enjoined.

6 28. To the extent Defendants' Infringing Products, without more, do not directly infringe
 7 at least claim 11 of the '175 Patent, at least as of the filing of this Complaint, Defendants contribute
 8 to infringement of the same under 35 U.S.C. § 271(c) inasmuch as the Infringing Products offered for
 9 sale and sold by Defendants are each a component of a patented machine or an apparatus used in
 10 practicing a patented process, constituting a material part of SRI's invention, knowing the same to be
 11 especially made or especially adapted for use in infringement of the '175 Patent. For example, as set
 12 forth above, the Test Studio, when used in its normal and intended usage (pursuant to the instructions
 13 set forth on Defendants' websites), infringes claim 11 of the '175 Patent. *See supra*, ¶ 26.

14 29. Defendants will, on information and belief, continue to irreparably harm SRI unless
 15 enjoined.

16 30. Defendants actively encourage their customers to use Defendants' Infringing Products
 17 in an infringing manner. For example, Defendants' website is replete with written directions,
 18 screenshots, and videos instructing users on how to use the Infringing Products in an infringing
 19 manner. For example, as set forth above, Defendants' website regarding the Test Studio specifically
 20 instructs users of the Infringing Products how to infringe claim 11 of the '175 patent. *See supra*,
 21 ¶ 26. Defendants' website also touts the identities of customers who use the Infringing Products,
 22 each of whom is a direct infringer inasmuch as they use the Infringing Products in the infringing
 23 manner as instructed by Defendants:

24 //

25 //

26 //

27 //

28

Mark Judson
SOFTWARE DEVELOPER,
EBSCO INDUSTRIES INC.

“ After using the trial for a week we increased our automated testing from 5% to 30%. Test Studio was so easy to incorporate into our existing testing environment! Tests are stable and easy to run and understand.

RELATED PRODUCTS: [Functional Testing - SILVERLIGHT](#) [Test Studio](#)

Ronak Samantray
SOFTWARE DEVELOPER,
NOWFLOATS TECHNOLOGIES

“ Love the controls! They make my life so simple. The best part is the product is indeed a WYSIWIG :) - it delivers what it promises.

RELATED PRODUCTS: [Telerik UI for Windows 8](#) [Test Studio](#) [TeamPulse](#)

Amit Deshpande
TECHNICAL LEADER,
ITRA

“ Recently I downloaded Test Studio. Coming from a development background, I know how important is it to completely test any application before delivering it to the client. I was really amazed at the robustness of Test Studio. I can perform Functional, Performance, Load, etc. in one product. Earlier we were focusing on many products out there for each type of testing, however now its just one package .. Just Love it ..

RELATED PRODUCTS: [Test Studio](#)

31. Upon information and belief, and particularly by way of the detailed documentation instructing users on how to use the Infringing Products in an infringing manner (*see supra*, ¶¶ 26, 30), Defendants have encouraged this infringement with knowledge of the ‘175 Patent and with a specific intent to cause their customers and distributors to infringe.

32. Defendants will, on information and belief, continue to induce infringement of the ‘175 Patent unless enjoined.

33. Pursuant to 35 U.S.C. § 284, SRI is entitled to damages adequate to compensate for the infringement but in no event less than a reasonable royalty.

34. Defendants’ infringement has been and is willful and, pursuant to 35 U.S.C. § 284, SRI is entitled to treble damages. Defendants’ willful infringement is based at least on Defendants’ knowledge of SRI, its products, and its patents since at least as early as the filing of this Complaint. Defendants have either willfully and wantonly infringed the ‘175 Patent or have recklessly avoided knowledge of their own infringement, even when faced with knowledge of SRI’s own products and the Patents-in-Suit.

35. This case is “exceptional” within the meaning of 35 U.S.C. § 285, and SRI is entitled to an award of attorneys’ fees.

COUNT II – INFRINGEMENT OF THE ‘271 PATENT

36. SRI re-alleges and incorporates the allegations of the preceding paragraphs of this Complaint as if fully set forth herein.

37. SRI is the assignee and owner of all right, title, and interest in and to the ‘271 Patent, which was issued on December 4, 2012. A true and correct copy of the ‘271 Patent is attached hereto as Exhibit B.

38. The ‘271 Patent addresses an invention for testing websites. This disclosed innovation tests many facets of the website’s experience and operation, including by providing novel approaches to creating, storing, and executing test scripts using website elements as opposed to the previously disclosed use of recording test scripts based upon user actions only.

39. SRI has the exclusive right to make, use, sell, and offer to sell any product embodying the ‘271 Patent throughout the United States, and to import any product embodying the ‘271 Patent into the United States.

40. SRI has commercially exploited the ‘271 Patent by making, marketing, selling, and using products covered by the ‘271 Patent, including its popular eValid™ software products. SRI continues to commercially exploit the ‘271 Patent through the present, at least by continuing to provide maintenance and support to users of its popular eValid™ software products.

41. Defendants have had knowledge of the ‘271 Patent, SRI, and SRI’s products embodying the inventions claimed in the Patents-in-Suit since at least as early as the filing of this Complaint.

42. At all relevant times, SRI provided public notice of the ‘271 Patent at least by properly marking its products and its website pursuant to 35 U.S.C. § 287(a).

43. Defendants have been, and are currently, directly infringing at least claim 1 of the ‘271 Patent in violation of 35 U.S.C. § 271(a), literally or under the doctrine of equivalents, by making, using, selling, offering for sale, and/or importing into the United States Defendants’ Infringing Products, which, as set forth in documentation available on Defendants’ website, comprise the non-transitory computer readable media disclosed in the ‘271 Patent—both as maintained in Defendants’ files and as made accessible to its users to whom Defendants offer and sell the Infringing

1 Products—including at least computer program code stored therein for providing a test-enabled
 2 browser for testing a website residing on a network (for example, “Progress expertise in web
 3 technologies and leadership in UI component development is leveraged through our automated
 4 software testing tool: Progress Test Studio, to assist users in automating even the most complex web
 5 testing scenarios.”) (<https://www.telerik.com/teststudio/html-testing>); the Test Studio as used with a
 6 web browser is a test-enabled web browser that can be used for “Web applications testing”
 7 (<https://www.telerik.com/teststudio/html-testing>; see also <https://docs.telerik.com/teststudio/welcome>
 8 (“Test Studio functional testing is a comprehensive yet cost-effective automated testing suite.”)); the
 9 website necessarily resides on a network and the Test Studio utilizes any number of browsers as its
 10 “test-enabled browser” ([https://docs.telerik.com/teststudio/general-information/test-
 11 recording/overview](https://docs.telerik.com/teststudio/general-information/test-recording/overview); <https://www.telerik.com/teststudio/html-testing>); computer program code for
 12 interfacing with web browsing components, the web browsing components including DOM access
 13 methods of the web browsing components (for example, the Test Studio allows a user to browse the
 14 web via common web browsing activities, including navigating to a website and firing events such as
 15 clicking on a button) ([https://docs.telerik.com/teststudio/general-information/test-
 16 recording/overview](https://docs.telerik.com/teststudio/general-information/test-recording/overview)); the Test Studio interrogates the DOM to identify and extract relevant
 17 information regarding at least the page elements germane to the script, including each such element’s
 18 index and value, and stores those details in the test script ([https://docs.telerik.com/teststudio/general-
 19 information/test-results/analyze-test-list-results](https://docs.telerik.com/teststudio/general-information/test-results/analyze-test-list-results)); these page elements are located based on their DOM
 20 indexes (<https://docs.telerik.com/teststudio/features/recorder/dom-explorer>;
 21 [https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-
 22 wtc/element-identification-wtc/finding-page-elements](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements); [https://docs.telerik.com/teststudio/testing-
 23 framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-
 24 elements](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements); <https://docs.telerik.com/teststudio/general-information/configure-your-browser/edge>);
 25 computer program code for rendering and examining at least one webpage of the website so as to at
 26 least extract details of organization and structure of elements of the webpage, and store such details
 27 of the webpage in a recorded script, such as recorded scripts generated through the testing component
 28 of Defendants’ Infringing Products (for example, the Test Studio allows for the creation of test scripts

1 to test websites by recording a user's interactions with the webpage in question and allowing the user
 2 to play back those test scripts ([https://docs.telerik.com/teststudio/getting-started/first-project#test-](https://docs.telerik.com/teststudio/getting-started/first-project#test-recording)
 3 [recording; https://docs.telerik.com/teststudio/general-information/test-execution/quick-execution](https://docs.telerik.com/teststudio/general-information/test-execution/quick-execution));
 4 the Test Studio interrogates the DOM to identify and extract relevant information regarding at least
 5 the page elements germane to the script, including each such element's index and value, and stores
 6 those details in the test script ([https://docs.telerik.com/teststudio/general-information/test-](https://docs.telerik.com/teststudio/general-information/test-results/analyze-test-list-results)
 7 [results/analyze-test-list-results](https://docs.telerik.com/teststudio/general-information/test-results/analyze-test-list-results)); the Test Studio uses explicit wait commands that "wait for the
 8 comparison to be true before proceeding"—in order to perform such a validation, the Test Studio
 9 must necessarily store facts about the webpage being rendered, i.e., the expected condition to be
 10 checked for during validation (<https://docs.telerik.com/teststudio/features/recorder/verifications/wait>;
 11 <https://www.telerik.com/blogs/explicit-waits-with-ajax-combobox-example>); these page elements are
 12 located based on their DOM indexes ([https://docs.telerik.com/teststudio/features/recorder/dom-](https://docs.telerik.com/teststudio/features/recorder/dom-explorer)
 13 [explorer; https://docs.telerik.com/teststudio/features/recorder/dom-](https://docs.telerik.com/teststudio/features/recorder/dom-explorer)
 14 [explorer; https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements)
 15 [topics-wtc/element-identification-wtc/finding-page-elements](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements);
 16 <https://docs.telerik.com/teststudio/general-information/configure-your-browser/edge>); computer
 17 program code for selecting a validation test to be performed (for example, the Test Studio allows for
 18 the creation of test scripts to test websites by recording a user's interactions with the webpage in
 19 question and allowing the user to play back those test scripts
 20 (<https://docs.telerik.com/teststudio/getting-started/first-project#test-recording>;
 21 <https://docs.telerik.com/teststudio/general-information/test-execution/quick-execution>)); the Test
 22 Studio uses explicit wait commands to search for the expected elements against which it validates the
 23 webpage being rendered (<https://docs.telerik.com/teststudio/features/recorder/verifications/wait>;
 24 <https://www.telerik.com/blogs/explicit-waits-with-ajax-combobox-example>); and computer program
 25 code for performing the validation test using at least one of the DOM access methods of the web
 26 browsing components, wherein during the validation test, the at least one webpage is newly rendered
 27 and details of organization and structure of elements for the at least one webpage as newly rendered
 28 are accessed via the at least one of the DOM access methods and compared to the stored details in the
 recorded script (for example, the Test Studio allows for the creation of test scripts to test websites by

1 recording a user's interactions with the webpage in question and allowing the user to select and play
 2 back those test scripts (<https://docs.telerik.com/teststudio/getting-started/first-project#test-recording>;
 3 <https://docs.telerik.com/teststudio/general-information/test-execution/quick-execution>)); the Test
 4 Studio interrogates the DOM to identify and extract relevant information regarding at least the page
 5 elements germane to the script, including each such element's index and value, and stores those
 6 details in the test script ([https://docs.telerik.com/teststudio/general-information/test-results/analyze-](https://docs.telerik.com/teststudio/general-information/test-results/analyze-test-list-results)
 7 [test-list-results](https://docs.telerik.com/teststudio/general-information/test-results/analyze-test-list-results)); the Test Studio uses explicit wait commands to search for the expected elements
 8 against which it validates the webpage being rendered
 9 (<https://docs.telerik.com/teststudio/features/recorder/verifications/wait>;
 10 <https://www.telerik.com/blogs/explicit-waits-with-ajax-combobox-example>); and these page
 11 elements are located based on their DOM indexes
 12 ((<https://docs.telerik.com/teststudio/features/recorder/dom-explorer>;
 13 [https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements)
 14 [wtc/element-identification-wtc/finding-page-elements](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements); [https://docs.telerik.com/teststudio/testing-](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements)
 15 [framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements)
 16 [elements](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements); <https://docs.telerik.com/teststudio/general-information/configure-your-browser/edge>), as
 17 disclosed in the '271 Patent.

18 44. Defendants will, on information and belief, continue to directly infringe the '271
 19 Patent unless enjoined.

20 45. To the extent Defendants' Infringing Products, without more, do not directly infringe
 21 at least claim 1 of the '271 Patent, at least as of the filing of this Complaint, Defendants contribute to
 22 infringement of the same under 35 U.S.C. § 271(c) inasmuch as the Infringing Products offered for
 23 sale and sold by Defendants are each a component of a patented machine or an apparatus used in
 24 practicing a patented process, constituting a material part of SRI's invention, knowing the same to be
 25 especially made or especially adapted for use in infringement of the '271 Patent. For example, as set
 26 forth above, the Test Studio, when used in its normal and intended usage (pursuant to the instructions
 27 set forth on Defendants' website), infringes claim 1 of the '271 Patent. *See supra*, ¶ 43.

46. Defendants will, on information and belief, continue to contribute to infringement of the '271 Patent unless enjoined.

47. Defendants actively encourage their customers to use Defendants' Infringing Products in an infringing manner. For example, Defendants' website is replete with written directions, screenshots, and videos instructing users on how to use the Infringing Products in an infringing manner. For example, as set forth above, Defendants' website regarding the Test Studio specifically instructs users of the Infringing Products how to infringe claim 1 of the '271 patent. *See supra*, ¶ 43. Defendants' website also touts the identities of customers who use the Infringing Products, each of whom is a direct infringer inasmuch as they use the Infringing Products in the infringing manner as instructed by Defendants:

Mark Judson

SOFTWARE DEVELOPER,
EBSCO INDUSTRIES INC.



After using the trial for a week we increased our automated testing from 5% to 30%. Test Studio was so easy to incorporate into our existing testing environment! Tests are stable and easy to run and understand.

RELATED PRODUCTS:

Functional Testing - SILVERLIGHT

Test Studio

Ronak Samantray

SOFTWARE DEVELOPER,
NOWFLOATS TECHNOLOGIES



Love the controls! They make my life so simple. The best part is the product is indeed a WYSIWIG :) - it delivers what it promises.

RELATED PRODUCTS:

Telerik UI for Windows 8

Test Studio

TeamPulse

Amit Deshpande

TECHNICAL LEADER,
ITRA



Recently I downloaded Test Studio. Coming from a development background, I know how important is it to completely test any application before delivering it to the client. I was really amazed at the robustness of Test Studio. I can perform Functional, Performance, Load, etc. in one product. Earlier we were focusing on many products out there for each type of testing, however now its just one package .. Just Love it ..

RELATED PRODUCTS:

Test Studio

48. Upon information and belief, and particularly by way of the detailed documentation instructing users on how to use the Infringing Products in an infringing manner (*see supra*, ¶¶ 43, 47), Defendants have encouraged this infringement with knowledge of the '271 Patent and with a specific intent to cause their customers and distributors to infringe.

49. Defendants' acts at least as of the filing of this Complaint thus constitute active inducement of patent infringement in violation of 35 U.S.C. § 271(b).

50. Defendants will, on information and belief, continue to induce infringement of the '271 Patent unless enjoined.

51. Defendants' direct infringement, contributory infringement, and inducement of infringement have irreparably harmed SRI.

52. Defendants will, on information and belief, continue to irreparably harm SRI unless enjoined.

53. Pursuant to 35 U.S.C. § 284, SRI is entitled to damages adequate to compensate for the infringement but in no event less than a reasonable royalty.

54. Defendants' infringement has been and is willful and, pursuant to 35 U.S.C. § 284, SRI is entitled to treble damages. Defendants' willful infringement is based at least on Defendants' knowledge of SRI, its products, and its patents since at least as early as the filing of this Complaint. Defendants have either willfully and wantonly infringed the '271 Patent or have recklessly avoided knowledge of their own infringement, even when faced with knowledge of SRI's own products and the Patents-in-Suit.

55. This case is "exceptional" within the meaning of 35 U.S.C. § 285, and SRI is entitled to an award of attorneys' fees.

COUNT III – INFRINGEMENT OF THE '890 PATENT

56. SRI re-alleges and incorporates the allegations of the preceding paragraphs of this Complaint as if fully set forth herein.

57. SRI is the assignee and owner of all right, title, and interest in and to the '890 Patent, which was issued on March 5, 2013. A true and correct copy of the '890 Patent is attached hereto as Exhibit C.

58. The '890 Patent addresses an invention for testing websites. The disclosed innovation tests many facets of the website's experience and operation, including by providing novel approaches to creating, storing, and executing test scripts capable of accurately testing Asynchronous Javascript and XML ("AJAX") webpage elements.

59. SRI has the exclusive right to make, use, sell, and offer to sell any product embodying the '890 Patent throughout the United States, and to import any product embodying the '890 Patent into the United States.

60. SRI has commercially exploited the '890 Patent by making, marketing, selling, and using products covered by the '890 Patent, including its popular eValid™ software products. SRI continues to commercially exploit the '890 Patent through the present, at least by continuing to provide maintenance and support to users of its popular eValid™ software products.

61. Defendants have had knowledge of the '890 Patent, SRI, and SRI's products embodying the inventions claimed in the Patents-in-Suit since at least as early as the filing of this Complaint.

62. At all relevant times, SRI provided public notice of the '890 Patent by properly marking its products and its website pursuant to 35 U.S.C. § 287(a). Defendants have been, and are currently, directly infringing at least claim 1 of the '890 Patent in violation of 35 U.S.C. § 271(a), literally or under the doctrine of equivalents, by making, using, selling, offering for sale, and/or importing into the United States Defendants' Infringing Products, which, as set forth in documentation available on Defendants' website, comprise the non-transitory computer readable media disclosed in the '890 Patent—both as maintained in Defendants' files and as made accessible to its users to whom Defendants offer and sell the Infringing Products—including at least computer program code stored therein for providing a test-enabled web browser, said medium comprising computer program code for providing web browsing capabilities (for example, "Progress expertise in web technologies and leadership in UI component development is leveraged through our automated software testing tool: Progress Test Studio, to assist users in automating even the most complex web testing scenarios.") (<https://www.telerik.com/teststudio/html-testing>); the Test Studio as used with a web browser is a test-enabled web browser that can be used for "Web applications testing" (<https://www.telerik.com/teststudio/html-testing>; see also <https://docs.telerik.com/teststudio/welcome> ("Test Studio functional testing is a comprehensive yet cost-effective automated testing suite.")); the Test Studio utilizes any number of browsers as its "test-enabled browser" (<https://docs.telerik.com/teststudio/general-information/test-recording/overview>;

<https://www.telerik.com/teststudio/html-testing>) and allows a user to browse the web via common web browsing activities, including navigating to a website and firing events such as clicking on a button (<https://docs.telerik.com/teststudio/features/recorder/overview>; <https://docs.telerik.com/teststudio/general-information/test-recording/overview>; <https://docs.telerik.com/teststudio/getting-started/first-project#test-recording>); computer program code for testing capabilities of a website hosted by a server and accessible to the computer via a network wherein the computer program code for testing capabilities of the website includes at least computer program code configured to receive a synchronization check from a user using the test enabled browser, to insert the synchronization check into a test script for testing at least one webpage of the website (for example, the Test Studio allows for the creation of test scripts to test websites by recording a user's interactions with the webpage in question and allowing the user to play back those test scripts (<https://docs.telerik.com/teststudio/getting-started/first-project#test-recording>; <https://docs.telerik.com/teststudio/general-information/test-execution/quick-execution>)); the Test Studio interrogates the DOM to identify and extract relevant information regarding at least the page elements germane to the script, including each such element's index and value, and stores those details in the test script (<https://docs.telerik.com/teststudio/general-information/test-results/analyze-test-list-results>); the Test Studio uses explicit wait commands that "wait for the comparison to be true before proceeding"—in order to perform such a validation, the Test Studio must necessarily store facts about the webpage being rendered, i.e., the expected condition to be checked for during validation (<https://docs.telerik.com/teststudio/features/recorder/verifications/wait>; <https://www.telerik.com/blogs/explicit-waits-with-ajax-combobox-example>); these page elements are located based on their DOM indexes (<https://docs.telerik.com/teststudio/features/recorder/dom-explorer>; <https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements>; <https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements>; <https://docs.telerik.com/teststudio/general-information/configure-your-browser/edge>); and the Test Studio allows for the testing of content dynamically generated by AJAX programming including using, for example, its various wait

1 commands or similar technologies or other related functions to synchronize playback and allow for
 2 testing of content dynamically generated by AJAX programming
 3 (<https://docs.telerik.com/teststudio/features/recorder/verifications/wait>;
 4 <https://www.telerik.com/blogs/explicit-waits-with-ajax-combobox-example>); the test script being
 5 separate from the at least one webpage being tested (for example, the Test Studio stores and accesses
 6 test scripts separately from the webpage itself ([https://docs.telerik.com/teststudio/general-](https://docs.telerik.com/teststudio/general-information/test-execution/quick-execution)
 7 [information/test-execution/quick-execution](https://docs.telerik.com/teststudio/general-information/test-execution/quick-execution)), the at least one webpage being tested including AJAX
 8 programming, and to automatically synchronize playback of the test script using at least the
 9 synchronization check to maintain the test enabled browser's state with respect to the AJAX
 10 programming by means of the synchronization check in the test script to a Document Object Model
 11 (DOM) associated with the at least one webpage of the website (for example, the Test Studio allows
 12 for the creation of test scripts to test websites by recording a user's interactions with the webpage in
 13 question and allowing the user to play back those test scripts
 14 (<https://docs.telerik.com/teststudio/getting-started/first-project#test-recording>;
 15 <https://docs.telerik.com/teststudio/general-information/test-execution/quick-execution>)); the Test
 16 Studio interrogates the DOM to identify and extract relevant information regarding at least the page
 17 elements germane to the script, including each such element's index and value, and stores those
 18 details in the test script ([https://docs.telerik.com/teststudio/general-information/test-results/analyze-](https://docs.telerik.com/teststudio/general-information/test-results/analyze-test-list-results)
 19 [test-list-results](https://docs.telerik.com/teststudio/general-information/test-results/analyze-test-list-results)); the Test Studio uses explicit wait commands to search for the expected elements
 20 against which it validates the webpage being rendered
 21 (<https://docs.telerik.com/teststudio/features/recorder/verifications/wait>;
 22 <https://www.telerik.com/blogs/explicit-waits-with-ajax-combobox-example>); these page elements are
 23 located based on their DOM indexes ([https://docs.telerik.com/teststudio/features/recorder/dom-](https://docs.telerik.com/teststudio/features/recorder/dom-explorer)
 24 [explorer](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements); [https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements)
 25 [topics-wtc/element-identification-wtc/finding-page-elements](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements);
 26 [https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements)
 27 [wtc/element-identification-wtc/finding-page-elements](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements); [https://docs.telerik.com/teststudio/general-](https://docs.telerik.com/teststudio/general-information/configure-your-browser/edge)
 28 [information/configure-your-browser/edge](https://docs.telerik.com/teststudio/general-information/configure-your-browser/edge)); and the Test Studio allows for the testing of content

1 dynamically generated by AJAX programming using, for example, its various wait commands to
 2 synchronize playback and allow for testing of content dynamically generated by AJAX programming
 3 (<https://docs.telerik.com/teststudio/features/recorder/verifications/wait>;
 4 <https://www.telerik.com/blogs/explicit-waits-with-ajax-combobox-example>); wherein the
 5 synchronization check in the test script and web browsing activities provided by the web browsing
 6 capabilities are able to separately access the DOM associated with the at least one webpage of the
 7 website (for example, the Test Studio stores and accesses test scripts separately from the webpage
 8 itself (<https://docs.telerik.com/teststudio/general-information/test-execution/quick-execution>));
 9 wherein the synchronization check is inserted into the test script as at least one command, and the at
 10 least one command operates, when executed, to: find a current index of at least one DOM element of
 11 the at least one webpage based on a specified property name and/or property value; and (i) submit a
 12 named event to the at least one DOM element of the at least one webpage having the current index, or
 13 (ii) insert or verify a value in the at least one DOM element of the at least one webpage having the
 14 current index (for example, the Test Studio allows for the creation of test scripts to test websites by
 15 recording a user's interactions with the webpage in question and allowing the user to play back those
 16 test scripts (<https://docs.telerik.com/teststudio/getting-started/first-project#test-recording>;
 17 <https://docs.telerik.com/teststudio/general-information/test-execution/quick-execution>)); the Test
 18 Studio interrogates the DOM to identify and extract relevant information regarding at least the page
 19 elements germane to the script, including each such element's index and value, and stores those
 20 details in the test script ([https://docs.telerik.com/teststudio/general-information/test-results/analyze-
 21 test-list-results](https://docs.telerik.com/teststudio/general-information/test-results/analyze-test-list-results)); the Test Studio uses explicit wait commands that "wait for the comparison to be true
 22 before proceeding"—in order to perform such a validation, the Test Studio must necessarily store
 23 facts about the webpage being rendered, i.e., the expected condition to be checked for during
 24 validation (<https://docs.telerik.com/teststudio/features/recorder/verifications/wait>;
 25 <https://www.telerik.com/blogs/explicit-waits-with-ajax-combobox-example>); these page elements are
 26 located based on their DOM indexes ([https://docs.telerik.com/teststudio/features/recorder/dom-
 27 explorer](https://docs.telerik.com/teststudio/features/recorder/dom-explorer); [https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-
 28 topics-wtc/element-identification-wtc/finding-page-elements](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements));

1 [https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements)
 2 [wtc/element-identification-wtc/finding-page-elements](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements); [https://docs.telerik.com/teststudio/general-](https://docs.telerik.com/teststudio/general-information/configure-your-browser/edge)
 3 [information/configure-your-browser/edge](https://docs.telerik.com/teststudio/general-information/configure-your-browser/edge)); and the Test Studio allows for the testing of content
 4 dynamically generated by AJAX programming using, for example, its various wait commands to
 5 synchronize playback and allow for testing of content dynamically generated by AJAX programming
 6 (<https://docs.telerik.com/teststudio/features/recorder/verifications/wait>;
 7 <https://www.telerik.com/blogs/explicit-waits-with-ajax-combobox-example>), as disclosed in the '890
 8 Patent.

9 63. Defendants will, on information and belief, continue to directly infringe the '890
 10 Patent unless enjoined.

11 64. To the extent Defendants' Infringing Products, without more, do not directly infringe
 12 at least claim 1 of the '890 Patent, at least as of the filing of this Complaint, Defendants contribute to
 13 infringement of the same under 35 U.S.C. § 271(c) inasmuch as the Infringing Products offered for
 14 sale and sold by Defendants are each a component of a patented machine or an apparatus used in
 15 practicing a patented process, constituting a material part of SRI's invention, knowing the same to be
 16 especially made or especially adapted for use in infringement of the '890 Patent. For example, the
 17 Test Studio, when used in its normal and intended usage (pursuant to the instructions set forth on
 18 Defendants' website) infringes claim 1 of the '890 Patent. *See supra*, ¶ 62.

19 65. Defendants will, on information and belief, continue to contribute to infringement of
 20 the '890 Patent unless enjoined.

21 66. Defendants actively encourage their customer to use Defendants' Infringing Products
 22 in an infringing manner. For example, Defendants' website is replete with written directions,
 23 screenshots, and videos instructing users on how to use the Infringing Products in an infringing
 24 manner. For example, as set forth above, Defendants' website regarding the Test Studio specifically
 25 instructs users of the Infringing Products how to infringe claim 1 of the '890 patent. *See supra*, ¶ 62.
 26 Defendants' website also touts the identities of customers who use the Infringing Products, each of
 27 whom is a direct infringer inasmuch as they use the Infringing Products in the infringing manner as
 28 instructed by Defendants:

Mark Judson
SOFTWARE DEVELOPER,
EBSCO INDUSTRIES INC.

“ After using the trial for a week we increased our automated testing from 5% to 30%. Test Studio was so easy to incorporate into our existing testing environment! Tests are stable and easy to run and understand.

RELATED PRODUCTS: [Functional Testing - SILVERLIGHT](#) [Test Studio](#)

Ronak Samantray
SOFTWARE DEVELOPER,
NOWFLOATS TECHNOLOGIES

“ Love the controls! They make my life so simple. The best part is the product is indeed a WYSIWIG :) - it delivers what it promises.

RELATED PRODUCTS: [Telerik UI for Windows 8](#) [Test Studio](#) [TeamPulse](#)

Amit Deshpande
TECHNICAL LEADER,
ITRA

“ Recently I downloaded Test Studio. Coming from a development background, I know how important is it to completely test any application before delivering it to the client. I was really amazed at the robustness of Test Studio. I can perform Functional, Performance, Load, etc. in one product. Earlier we were focusing on many products out there for each type of testing, however now its just one package .. Just Love it ..

RELATED PRODUCTS: [Test Studio](#)

67. Upon information and belief, and particularly by way of the detailed documentation instructing users on how to use the Infringing Products in an infringing manner (*see supra*, ¶¶ 62, 66), Defendants have encouraged this infringement with knowledge of the '890 Patent and with a specific intent to cause their customers and distributors to infringe.

68. Defendants' acts at least as of the filing of this Complaint thus constitute active inducement of patent infringement in violation of 35 U.S.C. § 271(b).

69. Defendants will, on information and belief, continue to induce infringement of the '890 Patent unless enjoined.

70. Defendants' direct infringement, contributory infringement, and inducement of infringement have irreparably harmed SRI.

71. Defendants will, on information and belief, continue to irreparably harm SRI unless enjoined.

72. Pursuant to 35 U.S.C. § 284, SRI is entitled to damages adequate to compensate for the infringement but in no event less than a reasonable royalty.

73. Defendants' infringement has been and is willful and, pursuant to 35 U.S.C. § 284, SRI is entitled to treble damages. Defendants' willful infringement is based at least on Defendants'

1 knowledge of SRI, its products, and its patents since at least as early as the filing of this Complaint.
2 Defendants have either willfully and wantonly infringed the '890 Patent or have recklessly avoided
3 knowledge of their own infringement, even when faced with knowledge of SRI's own products and
4 the Patents-in-Suit.

5 74. This case is "exceptional" within the meaning of 35 U.S.C. § 285, and SRI is entitled
6 to an award of attorneys' fees.

7 **COUNT IV – INFRINGEMENT OF THE '585 PATENT**

8 75. SRI re-alleges and incorporates the allegations of the preceding paragraphs of this
9 Complaint as if fully set forth herein.

10 76. SRI is the assignee and owner of all right, title, and interest in and to the '585 Patent,
11 which was issued on July 23, 2013. A true and correct copy of the '585 Patent is attached hereto as
12 Exhibit D.

13
14 77. The '585 Patent addresses an invention for testing websites. The disclosed innovation
15 tests many facets of the website's experience and operation, including by providing novel approaches
16 to creating, storing, and executing test scripts capable of accurately testing AJAX webpage elements.

17 78. SRI has the exclusive right to make, use, sell, and offer to sell any product embodying
18 the '585 Patent throughout the United States, and to import any product embodying the '585 Patent
19 into the United States.

20 79. SRI has commercially exploited the '585 Patent by making, marketing, selling, and
21 using products covered by the '585 Patent, including its popular eValid™ software products. SRI
22 continues to commercially exploit the '585 Patent through the present, at least by continuing to
23 provide maintenance and support to users of its popular eValid™ software products.

24 80. Defendants have had knowledge of the '585 Patent, SRI, and SRI's products
25 embodying the inventions claimed in the Patents-in-Suit since at least as early as the filing of this
26 Complaint.

27 81. At all relevant times, SRI provided public notice of the '585 Patent by properly
28 marking its products and its website pursuant to 35 U.S.C. § 287(a).

82. Defendants have been, and are currently, directly infringing at least claim 1 of the '585 Patent in violation of 35 U.S.C. § 271(a), literally or under the doctrine of equivalents, by making, using, selling, offering for sale, and/or importing into the United States Defendants' Infringing Products, which, as set forth in documentation available on Defendants' website, comprise the non-transitory computer readable media disclosed in the '585 Patent—both as maintained in Defendants' files and as made accessible to its users to whom Defendants offer and sell the Infringing Products—including at least computer program code for providing a test enabled web browser, said medium comprising computer program code for providing web browsing capabilities (for example, “Progress expertise in web technologies and leadership in UI component development is leveraged through our automated software testing tool: Progress Test Studio, to assist users in automating even the most complex web testing scenarios.”) (<https://www.telerik.com/teststudio/html-testing>); the Test Studio as used with a web browser is a test-enabled web browser that can be used for “Web applications testing” (<https://www.telerik.com/teststudio/html-testing>; see also <https://docs.telerik.com/teststudio/welcome> (“Test Studio functional testing is a comprehensive yet cost-effective automated testing suite.”); the Test Studio utilizes any number of browsers as its “test enabled browser” (<https://docs.telerik.com/teststudio/general-information/test-recording/overview>; <https://www.telerik.com/teststudio/html-testing>) and allows a user to browse the web via common web browsing activities, including navigating to a website and firing events such as clicking on a button (<https://docs.telerik.com/teststudio/features/recorder/overview>; <https://docs.telerik.com/teststudio/general-information/test-recording/overview>; <https://docs.telerik.com/teststudio/getting-started/first-project#test-recording>); computer program code for testing capabilities of a website hosted by a server and accessible to a computer via a network wherein the computer program code for testing capabilities of the website includes computer program code configured to receive a synchronization check from a user using the test enabled web browser, to insert the synchronization check into a test script for testing at least one webpage of the website (for example, the Test Studio allows for the creation of test scripts to test websites by recording a user's interactions with the webpage in question and allowing the user to play back those test scripts (<https://docs.telerik.com/teststudio/getting-started/first-project#test-recording>;

<https://docs.telerik.com/teststudio/general-information/test-execution/quick-execution>)); the Test Studio interrogates the DOM to identify and extract relevant information regarding at least the page elements germane to the script, including each such element's index and value, and stores those details in the test script (<https://docs.telerik.com/teststudio/general-information/test-results/analyze-test-list-results>); the Test Studio uses explicit wait commands that “wait for the comparison to be true before proceeding”—in order to perform such a validation, the Test Studio must necessarily store facts about the webpage being rendered, i.e., the expected condition to be checked for during validation (<https://docs.telerik.com/teststudio/features/recorder/verifications/wait>; <https://www.telerik.com/blogs/explicit-waits-with-ajax-combobox-example>); these page elements are located based on their DOM indexes (<https://docs.telerik.com/teststudio/features/recorder/dom-explorer>; <https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements>; <https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements>; <https://docs.telerik.com/teststudio/general-information/configure-your-browser/edge>); the Test Studio allows for the testing of content dynamically generated by AJAX programming including using, for example, its various wait commands or similar technologies or other related functions to synchronize playback and allow for testing of content dynamically generated by AJAX programming (<https://docs.telerik.com/teststudio/features/recorder/verifications/wait>; <https://www.telerik.com/blogs/explicit-waits-with-ajax-combobox-example>); the test script being separate from the at least one webpage being tested (for example, the Test Studio stores and accesses test scripts separately from the webpage itself (<https://docs.telerik.com/teststudio/general-information/test-execution/quick-execution>)), the at least one webpage being tested including AJAX programming, and to automatically synchronize playback of the test script using at least the synchronization check to maintain the test enabled browser's state with respect to the AJAX programming by means of the synchronization check in the test script to a DOM associated with the website (for example, the Test Studio allows for the creation of test scripts to test websites by recording a user's interactions with the webpage in question and allowing the user to play back those

1 test scripts (<https://docs.telerik.com/teststudio/getting-started/first-project#test-recording>;
 2 <https://docs.telerik.com/teststudio/general-information/test-execution/quick-execution>)); the Test
 3 Studio interrogates the DOM to identify and extract relevant information regarding at least the page
 4 elements germane to the script, including each such element's index and value, and stores those
 5 details in the test script ([https://docs.telerik.com/teststudio/general-information/test-results/analyze-
 6 test-list-results](https://docs.telerik.com/teststudio/general-information/test-results/analyze-test-list-results)); the Test Studio uses explicit wait commands that “wait for the comparison to be true
 7 before proceeding”—in order to perform such a validation, the Test Studio must necessarily store
 8 facts about the webpage being rendered, i.e., the expected condition to be checked for during
 9 validation (<https://docs.telerik.com/teststudio/features/recorder/verifications/wait>;
 10 <https://www.telerik.com/blogs/explicit-waits-with-ajax-combobox-example>); these page elements are
 11 located based on their DOM indexes ([https://docs.telerik.com/teststudio/features/recorder/dom-
 12 explorer](https://docs.telerik.com/teststudio/features/recorder/dom-explorer); [https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-
 13 topics-wtc/element-identification-wtc/finding-page-elements](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements);
 14 [https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-
 15 wtc/element-identification-wtc/finding-page-elements](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements); [https://docs.telerik.com/teststudio/general-
 16 information/configure-your-browser/edge](https://docs.telerik.com/teststudio/general-information/configure-your-browser/edge)); the Test Studio allows for the testing of content
 17 dynamically generated by AJAX programming including using, for example, its various wait
 18 commands or similar technologies or other related functions to synchronize playback and allow for
 19 testing of content dynamically generated by AJAX programming
 20 (<https://docs.telerik.com/teststudio/features/recorder/verifications/wait>;
 21 <https://www.telerik.com/blogs/explicit-waits-with-ajax-combobox-example>); wherein the
 22 synchronization check in the test script and web browsing activities provided by the web browsing
 23 capabilities are able to separately access the DOM associated with the at least one webpage of the
 24 website (for example, the Test Studio stores and accesses test scripts separately from the webpage
 25 itself (<https://docs.telerik.com/teststudio/general-information/test-execution/quick-execution>)); and
 26 wherein the synchronization check is inserted into the test script as at least one command, and the at
 27 least one command operates, when executed, to find a current index of at least one DOM element of
 28 the at least one webpage based on a specified property name and/or property value, and (i) submit a

1 named event to the at least one DOM element of the at least one webpage having the current index, or
 2 (ii) insert or verify a value in the at least one DOM element of the at least one webpage having the
 3 current index (for example, the Test Studio allows for the creation of test scripts to test websites by
 4 recording a user's interactions with the webpage in question and allowing the user to play back those
 5 test scripts (<https://docs.telerik.com/teststudio/getting-started/first-project#test-recording>;
 6 <https://docs.telerik.com/teststudio/general-information/test-execution/quick-execution>)); the Test
 7 Studio interrogates the DOM to identify and extract relevant information regarding at least the page
 8 elements germane to the script, including each such element's index and value, and stores those
 9 details in the test script ([https://docs.telerik.com/teststudio/general-information/test-results/analyze-](https://docs.telerik.com/teststudio/general-information/test-results/analyze-test-list-results)
 10 [test-list-results](https://docs.telerik.com/teststudio/general-information/test-results/analyze-test-list-results)); the Test Studio uses explicit wait commands that "wait for the comparison to be true
 11 before proceeding"—in order to perform such a validation, the Test Studio must necessarily store
 12 facts about the webpage being rendered, i.e., the expected condition to be checked for during
 13 validation (<https://docs.telerik.com/teststudio/features/recorder/verifications/wait>;
 14 <https://www.telerik.com/blogs/explicit-waits-with-ajax-combobox-example>); these page elements are
 15 located based on their DOM indexes ([https://docs.telerik.com/teststudio/features/recorder/dom-](https://docs.telerik.com/teststudio/features/recorder/dom-explorer)
 16 [explorer](https://docs.telerik.com/teststudio/features/recorder/dom-explorer); [https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements)
 17 [topics-wtc/element-identification-wtc/finding-page-elements](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements);
 18 [https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements)
 19 [wtc/element-identification-wtc/finding-page-elements](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements); [https://docs.telerik.com/teststudio/general-](https://docs.telerik.com/teststudio/general-information/configure-your-browser/edge)
 20 [information/configure-your-browser/edge](https://docs.telerik.com/teststudio/general-information/configure-your-browser/edge)); and the Test Studio allows for the testing of content
 21 dynamically generated by AJAX programming using, for example, its various wait commands to
 22 synchronize playback and allow for testing of content dynamically generated by AJAX programming
 23 (<https://docs.telerik.com/teststudio/features/recorder/verifications/wait>;
 24 <https://www.telerik.com/blogs/explicit-waits-with-ajax-combobox-example>), as disclosed in the '585
 25 Patent.

26 83. Defendants will, on information and belief, continue to directly infringe the '585
 27 Patent unless enjoined.

1 84. To the extent Defendants' Infringing Products, without more, do not directly infringe
2 at least claim 1 of the '585 Patent, at least as of the filing of this Complaint, Defendants contribute to
3 infringement of the same under 35 U.S.C. § 271(c) inasmuch as the Infringing Products offered for
4 sale and sold by Defendants are each a component of a patented machine or an apparatus used in
5 practicing a patented process, constituting a material part of SRI's invention, knowing the same to be
6 especially made or especially adapted for use in infringement of the '585 Patent. For example, as set
7 forth above, the Test Studio, when used in its normal and intended usage (pursuant to the instructions
8 set forth on Defendants' website) infringes claim 1 of the '585 Patent. *See supra*, ¶ 82.

9 85. Defendants will, on information and belief, continue to contribute to infringement of
10 the '585 Patent unless enjoined.

11 86. Defendants actively encourage their customer to use Defendants' Infringing Products
12 in an infringing manner. For example, Defendants' website is replete with written directions,
13 screenshots, and videos instructing users on how to use the Infringing Products in an infringing
14 manner. For example, as set forth above, Defendants' website regarding the Test Studio specifically
15 instructs users of the Infringing Products how to infringe claim 1 of the '585 patent. *See supra*, ¶ 82.
16 Defendants' website also touts the identities of customers who use the Infringing Products, each of
17 whom is a direct infringer inasmuch as they use the Infringing Products in the infringing manner as
18 instructed by Defendants:

Mark Judson
SOFTWARE DEVELOPER,
EBSCO INDUSTRIES INC.

“ After using the trial for a week we increased our automated testing from 5% to 30%. Test Studio was so easy to incorporate into our existing testing environment! Tests are stable and easy to run and understand.

RELATED PRODUCTS: [Functional Testing - SILVERLIGHT](#) [Test Studio](#)

Ronak Samantray
SOFTWARE DEVELOPER,
NOWFLOATS TECHNOLOGIES

“ Love the controls! They make my life so simple. The best part is the product is indeed a WYSIWIG :) - it delivers what it promises.

RELATED PRODUCTS: [Telerik UI for Windows 8](#) [Test Studio](#) [TeamPulse](#)

Amit Deshpande
TECHNICAL LEADER,
ITRA

“ Recently I downloaded Test Studio. Coming from a development background, I know how important is it to completely test any application before delivering it to the client. I was really amazed at the robustness of Test Studio. I can perform Functional, Performance, Load, etc. in one product. Earlier we were focusing on many products out there for each type of testing, however now its just one package .. Just Love it ..

RELATED PRODUCTS: [Test Studio](#)

87. Upon information and belief, and particularly by way of the detailed documentation instructing users on how to use the Infringing Products in an infringing manner (*see supra*, ¶¶ 82, 86), Defendants have encouraged this infringement with knowledge of the '585 Patent and with a specific intent to cause their customers and distributors to infringe.

88. Defendants' acts at least as of the filing of this Complaint thus constitute active inducement of patent infringement in violation of 35 U.S.C. § 271(b).

89. Defendants will, on information and belief, continue to induce infringement of the '585 Patent unless enjoined.

90. Defendants' direct infringement, contributory infringement, and inducement of infringement have irreparably harmed SRI.

91. Defendants will, on information and belief, continue to irreparably harm SRI unless enjoined.

92. Pursuant to 35 U.S.C. § 284, SRI is entitled to damages adequate to compensate for the infringement but in no event less than a reasonable royalty.

93. Defendants' infringement has been and is willful and, pursuant to 35 U.S.C. § 284, SRI is entitled to treble damages. Defendants' willful infringement is based at least on Defendants' knowledge of SRI, its products, and its patents since at least as early as the filing of this Complaint.

1 Defendants have either willfully and wantonly infringed the '585 Patent or have recklessly avoided
2 knowledge of their own infringement, even when faced with knowledge of SRI's own products and
3 the Patents-in-Suit.

4 94. This case is "exceptional" within the meaning of 35 U.S.C. § 285, and SRI is entitled
5 to an award of attorneys' fees.

6 **COUNT V – INFRINGEMENT OF THE '493 PATENT**

7 95. SRI re-alleges and incorporates the allegations of the preceding paragraphs of this
8 Complaint as if fully set forth herein.

9 96. SRI is the assignee and owner of all right, title, and interest in and to the '493 Patent,
10 which was issued on February 11, 2014. A true and correct copy of the '493 Patent is attached hereto
11 as Exhibit E.

12 97. The '493 Patent addresses an invention for testing websites. The disclosed innovation
13 tests many facets of the website's experience and operation, including by providing novel approaches
14 to creating, storing, and executing test scripts using website elements as opposed to the previously
15 disclosed use of recording test scripts based upon user actions only.

16 98. SRI has the exclusive right to make, use, sell, and offer to sell any product embodying
17 the '493 Patent throughout the United States, and to import any product embodying the '493 Patent
18 into the United States.

19 99. SRI has commercially exploited the '493 Patent by making, marketing, selling, and
20 using products covered by the '493 Patent, including its popular eValid™ software products. SRI
21 continues to commercially exploit the '493 Patent through the present, at least by continuing to
22 provide maintenance and support to users of its popular eValid™ software products.

23 100. Defendants have had knowledge of the '493 Patent, SRI, and SRI's products
24 embodying the inventions claimed in the Patents-in-Suit since at least as early as the filing of this
25 Complaint.

26 101. At all relevant times, SRI provided public notice of the '493 Patent by properly
27 marking its products and its website under 35 U.S.C. § 287(a).

Defendants have been, and are currently, directly infringing at least claim 1 of the '493 Patent in violation of 35 U.S.C. § 271(a), literally or under the doctrine of equivalents, by making, using, selling, offering for sale, and/or importing into the United States Defendants' Infringing Products, which, as set forth in documentation available on Defendants' website, comprise the non-transitory computer readable media disclosed in the '493 Patent—both as maintained in Defendants' files and as made accessible to its users to whom Defendants offer and sell the Infringing Products—including at least computer program code stored therein for providing a test-enabled browser for testing a website residing on a network (for example, "Progress expertise in web technologies and leadership in UI component development is leveraged through our automated software testing tool: Progress Test Studio, to assist users in automating even the most complex web testing scenarios.") (<https://www.telerik.com/teststudio/html-testing>); the Test Studio as used with a web browser is a test-enabled web browser that can be used for "Web applications testing" (<https://www.telerik.com/teststudio/html-testing>; see also <https://docs.telerik.com/teststudio/welcome> ("Test Studio functional testing is a comprehensive yet cost-effective automated testing suite."); the website necessarily resides on a network; and the Test Studio utilizes any number of browsers as its "test-enabled browser" (<https://docs.telerik.com/teststudio/general-information/test-recording/overview>; <https://www.telerik.com/teststudio/html-testing>), said medium comprising computer program code for interfacing with web browsing components, the web browsing components including DOM access methods, computer program code for accessing a website to be tested (for example the Test Studio allows a user to browse the web via common web browsing activities, including navigating to a website and firing events such as clicking on a button (<https://docs.telerik.com/teststudio/features/recorder/overview>; <https://docs.telerik.com/teststudio/general-information/test-recording/overview>; <https://docs.telerik.com/teststudio/getting-started/first-project#test-recording>)); the Test Studio interrogates the DOM to identify and extract relevant information regarding at least the page elements germane to the script, including each such element's index and value, and stores those details in the test script (<https://docs.telerik.com/teststudio/general-information/test-results/analyze-test-list-results>); these page elements are located based on their DOM indexes

(<https://docs.telerik.com/teststudio/features/recorder/dom-explorer>;
<https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements>; <https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements>; <https://docs.telerik.com/teststudio/general-information/configure-your-browser/edge>);
computer program code for rendering and examining at least one webpage of the website so as to
extract details of elements of the webpage, and store the details of the webpage in a recorded script,
such as recorded scripts generated through the testing component of the Infringing Products (for
example, the Test Studio allows for the creation of test scripts to test websites by recording a user's
interactions with the webpage in question and allowing the user to play back those test scripts
(<https://docs.telerik.com/teststudio/getting-started/first-project#test-recording>;
<https://docs.telerik.com/teststudio/general-information/test-execution/quick-execution>)); the Test
Studio interrogates the DOM to identify and extract relevant information regarding at least the page
elements germane to the script, including each such element's index and value, and stores those
details in the test script (<https://docs.telerik.com/teststudio/general-information/test-results/analyze-test-list-results>); the Test Studio uses explicit wait commands that "wait for the comparison to be true
before proceeding"—in order to perform such a validation, the Test Studio must necessarily store
facts about the webpage being rendered, i.e., the expected condition to be checked for during
validation (<https://docs.telerik.com/teststudio/features/recorder/verifications/wait>;
<https://www.telerik.com/blogs/explicit-waits-with-ajax-combobox-example>); these page elements are
located based on their DOM indexes (<https://docs.telerik.com/teststudio/features/recorder/dom-explorer>; <https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements>;
<https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements>; <https://docs.telerik.com/teststudio/general-information/configure-your-browser/edge>); computer program code for selecting a validation test to
be performed (for example, the Test Studio allows for the creation of test scripts to test websites by
recording a user's interactions with the webpage in question and allowing the user to play back those

1 test scripts (<https://docs.telerik.com/teststudio/getting-started/first-project#test-recording>;
 2 <https://docs.telerik.com/teststudio/general-information/test-execution/quick-execution>)); and the Test
 3 Studio uses explicit wait commands that “wait for the comparison to be true before proceeding”—in
 4 order to perform such a validation, the Test Studio must necessarily store facts about the webpage
 5 being rendered, i.e., the expected condition to be checked for during validation
 6 (<https://docs.telerik.com/teststudio/features/recorder/verifications/wait>;
 7 <https://www.telerik.com/blogs/explicit-waits-with-ajax-combobox-example>); and computer program
 8 code for performing the validation test using at least one of the DOM access methods of the web
 9 browsing components, wherein during the validation test, the at least one webpage is newly rendered
 10 and details of elements for the at least one webpage as newly rendered are accessed via the at least
 11 one of the DOM access methods and compared to the stored details in the recorded script (for
 12 example, the Test Studio allows for the creation of test scripts to test websites by recording a user’s
 13 interactions with the webpage in question and allowing the user to play back those test scripts
 14 (<https://docs.telerik.com/teststudio/getting-started/first-project#test-recording>;
 15 <https://docs.telerik.com/teststudio/general-information/test-execution/quick-execution>)); the Test
 16 Studio interrogates the DOM to identify and extract relevant information regarding at least the page
 17 elements germane to the script, including each such element’s index and value, and stores those
 18 details in the test script ([https://docs.telerik.com/teststudio/general-information/test-results/analyze-
 19 test-list-results](https://docs.telerik.com/teststudio/general-information/test-results/analyze-test-list-results)); the Test Studio uses explicit wait commands that “wait for the comparison to be true
 20 before proceeding”—in order to perform such a validation, the Test Studio must necessarily store
 21 facts about the webpage being rendered, i.e., the expected condition to be checked for during
 22 validation (<https://docs.telerik.com/teststudio/features/recorder/verifications/wait>;
 23 <https://www.telerik.com/blogs/explicit-waits-with-ajax-combobox-example>); and these page
 24 elements are located based on their DOM indexes
 25 (<https://docs.telerik.com/teststudio/features/recorder/dom-explorer>;
 26 [https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-
 27 wtc/element-identification-wtc/finding-page-elements](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements); [https://docs.telerik.com/teststudio/testing-
 28 framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-)

1 [elements; https://docs.telerik.com/teststudio/general-information/configure-your-browser/edge](https://docs.telerik.com/teststudio/general-information/configure-your-browser/edge)), as
2 disclosed in the '493 Patent.

3 102. Defendants will, on information and belief, continue to directly infringe the '493
4 Patent unless enjoined.

5 103. To the extent Defendants' Infringing Products, without more, do not directly infringe
6 at least claim 1 of the '493 Patent, at least as of the filing of this Complaint, Defendants contribute to
7 infringement of the same under 35 U.S.C. § 271(c) inasmuch as the Infringing Products offered for
8 sale and sold by Defendants are each a component of a patented machine or an apparatus used in
9 practicing a patented process, constituting a material part of SRI's invention, knowing the same to be
10 especially made or especially adapted for use in infringement of the '493 Patent. For example, as set
11 forth above, the Test Studio, when used in its normal and intended usage (pursuant to the instructions
12 set forth on Defendants' website) infringes claim 1 of the '493 Patent. *See supra*, ¶ 101.

13 104. Defendants will, on information and belief, continue to contribute to infringement of
14 the '493 Patent unless enjoined.

15 105. Defendants actively encourage their customer to use Defendants' Infringing Products
16 in an infringing manner. For example, Defendants' website is replete with written directions,
17 screenshots, and videos instructing users on how to use the Infringing Products in an infringing
18 manner. For example, as set forth above, Defendants' website regarding the Test Studio specifically
19 instructs users of the Infringing Products how to infringe claim 1 of the '493 patent. *See supra*,
20 ¶ 101. Defendants' website also touts the identities of customers who use the Infringing Products,
21 each of whom is a direct infringer inasmuch as they use the Infringing Products in the infringing
22 manner as instructed by Defendants:

Mark Judson
SOFTWARE DEVELOPER,
EBSCO INDUSTRIES INC.

“ After using the trial for a week we increased our automated testing from 5% to 30%. Test Studio was so easy to incorporate into our existing testing environment! Tests are stable and easy to run and understand.

RELATED PRODUCTS: [Functional Testing - SILVERLIGHT](#) [Test Studio](#)

Ronak Samantray
SOFTWARE DEVELOPER,
NOWFLOATS TECHNOLOGIES

“ Love the controls! They make my life so simple. The best part is the product is indeed a WYSIWIG :) - it delivers what it promises.

RELATED PRODUCTS: [Telerik UI for Windows 8](#) [Test Studio](#) [TeamPulse](#)

Amit Deshpande
TECHNICAL LEADER,
ITRA

“ Recently I downloaded Test Studio. Coming from a development background, I know how important is it to completely test any application before delivering it to the client. I was really amazed at the robustness of Test Studio. I can perform Functional, Performance, Load, etc. in one product. Earlier we were focusing on many products out there for each type of testing, however now its just one package .. Just Love it ..

RELATED PRODUCTS: [Test Studio](#)

106. Upon information and belief, and particularly by way of the detailed documentation instructing users on how to use the Infringing Products in an infringing manner (*see supra*, ¶¶ 101, 105), Defendants have encouraged this infringement with knowledge of the '493 Patent and with a specific intent to cause their customers and distributors to infringe.

107. Defendants' acts at least as of the filing of this Complaint thus constitute active inducement of patent infringement in violation of 35 U.S.C. § 271(b).

108. Defendants will, on information and belief, continue to induce infringement of the '493 Patent unless enjoined.

109. Defendants' direct infringement, contributory infringement, and inducement of infringement have irreparably harmed SRI.

110. Defendants will, on information and belief, continue to irreparably harm SRI unless enjoined.

111. Pursuant to 35 U.S.C. § 284, SRI is entitled to damages adequate to compensate for the infringement but in no event less than a reasonable royalty.

112. Defendants' infringement has been and is willful and, pursuant to 35 U.S.C. § 284, SRI is entitled to treble damages. Defendants' willful infringement is based at least on Defendants' knowledge of SRI, its products, and its patents since at least as early as the filing of this Complaint.

1 Defendants have either willfully and wantonly infringed the '493 Patent or have recklessly avoided
2 knowledge of their own infringement, even when faced with knowledge of SRI's own products and
3 patents.

4 113. This case is "exceptional" within the meaning of 35 U.S.C. § 285, and SRI is entitled
5 to an award of attorneys' fees.

6 **COUNT VI – INFRINGEMENT OF THE '491 PATENT**

7 114. SRI re-alleges and incorporates the allegations of the preceding paragraphs of this
8 Complaint as if fully set forth herein.

9 115. SRI is the assignee and owner of all right, title, and interest in and to the '491 Patent,
10 which was issued on March 17, 2015. A true and correct copy of the '491 Patent is attached hereto as
11 Exhibit F.

12 116. The '491 Patent addresses an invention for testing websites. The disclosed innovation
13 tests many facets of the website's experience and operation, including by providing novel approaches
14 to creating, storing, and executing test scripts using website elements as opposed to the previously
15 disclosed use of recording test scripts based upon user actions only.

16 117. SRI has the exclusive right to make, use, sell, and offer to sell any product embodying
17 the '491 Patent throughout the United States, and to import any product embodying the '491 Patent
18 into the United States.

19 118. SRI has commercially exploited the '491 Patent by making, marketing, selling, and
20 using products covered by the '491 Patent, including its popular eValid™ software products. SRI
21 continues to commercially exploit the '491 Patent through the present, at least by continuing to
22 provide maintenance and support to users of its popular eValid™ software products.

23 119. Defendants have had knowledge of the '491 Patent, SRI, and SRI's products
24 embodying the inventions claimed in the Patents-in-Suit since at least as early as the filing of this
25 Complaint.

26 120. At all relevant times, SRI provided public notice of the '491 Patent by properly
27 marking its products and its website pursuant to 35 U.S.C. § 287(a).

Defendants have been, and are currently, directly infringing at least claim 1 of the '491 Patent in violation of 35 U.S.C. § 271(a), literally or under the doctrine of equivalents, by making, using, selling, offering for sale, and/or importing into the United States Defendants' Infringing Products, which, as set forth in documentation available on Defendants' website, comprise the non-transitory computer readable media disclosed in the '491 Patent—both as maintained in Defendants' files and as made accessible to its users to whom Defendants offer and sell the Infringing Products—including at least computer program code for testing capabilities of a website hosted by a server and accessible to a computer via a network (for example, “Progress expertise in web technologies and leadership in UI component development is leveraged through our automated software testing tool: Progress Test Studio, to assist users in automating even the most complex web testing scenarios.”) (<https://www.telerik.com/teststudio/html-testing>); the Test Studio as used with a web browser is a test-enabled web browser that can be used for “Web applications testing” (<https://www.telerik.com/teststudio/html-testing>; see also <https://docs.telerik.com/teststudio/welcome> (“Test Studio functional testing is a comprehensive yet cost-effective automated testing suite.”); wherein the computer program code for testing capabilities of the website includes at least computer program code configured to have a synchronization check in a test script for testing at least one web page of the website, and to automatically synchronize playback of the test script using at least the synchronization check to maintain the test enabled browser's state by means of the synchronization check in the test script to a Document Object Model (DOM) associated with the at least one web page of the website, (for example, the Test Studio allows for the creation of test scripts to test websites by recording a user's interactions with the webpage in question and allowing the user to play back those test scripts (<https://docs.telerik.com/teststudio/getting-started/first-project#test-recording>; <https://docs.telerik.com/teststudio/general-information/test-execution/quick-execution>)); the Test Studio interrogates the DOM to identify and extract relevant information regarding at least the page elements germane to the script, including each such element's index and value, and stores those details in the test script (<https://docs.telerik.com/teststudio/general-information/test-results/analyze-test-list-results>); the Test Studio uses explicit wait commands that “wait for the comparison to be true before proceeding”—in order to perform such a validation, the Test Studio must necessarily store

facts about the webpage being rendered, i.e., the expected condition to be checked for during
 validation (<https://docs.telerik.com/teststudio/features/recorder/verifications/wait>;
<https://www.telerik.com/blogs/explicit-waits-with-ajax-combobox-example>); these page elements are
 located based on their DOM indexes (<https://docs.telerik.com/teststudio/features/recorder/dom-explorer>; <https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements>;
<https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements>; <https://docs.telerik.com/teststudio/general-information/configure-your-browser/edge>); and the Test Studio allows for the testing of content
 dynamically generated by AJAX programming using, for example, its various wait commands to
 synchronize playback and allow for testing of content dynamically generated by AJAX programming
 (<https://docs.telerik.com/teststudio/features/recorder/verifications/wait>;
<https://www.telerik.com/blogs/explicit-waits-with-ajax-combobox-example>); wherein the
 synchronization check operates, when executed, to: find a current index of at least one DOM element
 of the at least one web page based on a specified property name and/or property value; determine
 whether a property name and/or value is present in the at least one DOM element of the at least one
 web page having the current index; and after the current index is found and the property name and/or
 value is determined to be present, wait for the property name and/or value in the at least one DOM
 element of the at least one web page having the current index to be a particular name and/or value
 (for example, the Test Studio allows for the creation of test scripts to test websites by recording a
 user's interactions with the webpage in question and allowing the user to play back those test scripts,
 which are stored and accessed separately from the webpage itself
 (<https://docs.telerik.com/teststudio/general-information/test-execution/quick-execution>); the Test
 Studio interrogates the DOM to identify and extract relevant information regarding at least the page
 elements germane to the script, including each such element's index and value, and stores those
 details in the test script (<https://docs.telerik.com/teststudio/general-information/test-results/analyze-test-list-results>); the Test Studio uses explicit wait commands that "wait for the comparison to be true
 before proceeding"—in order to perform such a validation, the Test Studio must necessarily store

1 facts about the webpage being rendered, i.e., the expected condition to be checked for during
 2 validation (<https://docs.telerik.com/teststudio/features/recorder/verifications/wait>;
 3 <https://www.telerik.com/blogs/explicit-waits-with-ajax-combobox-example>); these page elements are
 4 located based on their DOM indexes ([https://docs.telerik.com/teststudio/features/recorder/dom-](https://docs.telerik.com/teststudio/features/recorder/dom-explorer)
 5 [explorer](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements); [https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements)
 6 [topics-wtc/element-identification-wtc/finding-page-elements](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements);
 7 [https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements)
 8 [wtc/element-identification-wtc/finding-page-elements](https://docs.telerik.com/teststudio/general-information/configure-your-browser/edge); [https://docs.telerik.com/teststudio/general-](https://docs.telerik.com/teststudio/general-information/configure-your-browser/edge)
 9 [information/configure-your-browser/edge](https://docs.telerik.com/teststudio/general-information/configure-your-browser/edge)); the Test Studio allows for the testing of content
 10 dynamically generated by AJAX programming using, for example, its various wait commands to
 11 synchronize playback and allow for testing of content dynamically generated by AJAX programming
 12 (<https://docs.telerik.com/teststudio/features/recorder/verifications/wait>;
 13 <https://www.telerik.com/blogs/explicit-waits-with-ajax-combobox-example>); wherein the computer
 14 program code configured to have the synchronization check is a separate programmatic process from
 15 the at least one web page of the website being tested; upon information and belief, the Test Studio
 16 synchronization processes (i.e., the Test Studio and/or browser plug-in) runs in a separate
 17 programmatic process from the web page of the website being tested (which runs in the web browser)
 18 (<https://docs.telerik.com/teststudio/general-information/test-execution/quick-execution>), as disclosed
 19 in the '491 Patent.

20 121. Defendants will, on information and belief, continue to directly infringe the '491
 21 Patent unless enjoined.

22 122. To the extent Defendants' Infringing Products, without more, do not directly infringe
 23 at least claim 1 of the '491 Patent, at least as of the filing of this Complaint, Defendants contribute to
 24 infringement of the same under 35 U.S.C. § 271(c) inasmuch as the Infringing Products offered for
 25 sale and sold by Defendants are each a component of a patented machine or an apparatus used in
 26 practicing a patented process, constituting a material part of SRI's invention, knowing the same to be
 27 especially made or especially adapted for use in infringement of the '491 Patent. For example, as set
 28

1 forth above, the Test Studio, when used in its normal and intended usage (pursuant to the instructions
2 set forth on Defendants' website), infringes claim 1 of the '491 Patent. *See supra*, ¶ 120.

3 123. Defendants will, on information and belief, continue to contribute to infringement of
4 the '491 Patent unless enjoined.

5 124. Defendants actively encourage their customer to use Defendants' Infringing Products
6 in an infringing manner. For example, Defendants' website is replete with written directions,
7 screenshots, and videos instructing users on how to use the Infringing Products in an infringing
8 manner. For example, as set forth above, Defendants' website regarding the Test Studio specifically
9 instructs users of the Infringing Products how to infringe claim 1 of the '491 patent. *See supra*,
10 ¶ 120. Defendants' website also touts the identities of customers who use the Infringing Products,
11 each of whom is a direct infringer inasmuch as they use the Infringing Products in the infringing
12 manner as instructed by Defendants:

13
14 **Mark Judson**

SOFTWARE DEVELOPER,
EBSCO INDUSTRIES INC.



After using the trial for a week we increased our automated testing from 5% to 30%. Test Studio was so easy to incorporate into our existing testing environment! Tests are stable and easy to run and understand.

RELATED PRODUCTS:

Functional Testing - SILVERLIGHT

Test Studio

17 **Ronak Samantray**

SOFTWARE DEVELOPER,
NOWFLOATS TECHNOLOGIES



Love the controls! They make my life so simple. The best part is the product is indeed a WYSIWIG :) - it delivers what it promises.

RELATED PRODUCTS:

Telerik UI for Windows 8

Test Studio

TeamPulse

20 **Amit Deshpande**

TECHNICAL LEADER,
ITRA



Recently I downloaded Test Studio. Coming from a development background, I know how important is it to completely test any application before delivering it to the client. I was really amazed at the robustness of Test Studio. I can perform Functional, Performance, Load, etc. in one product. Earlier we were focusing on many products out there for each type of testing, however now its just one package .. Just Love it ..

RELATED PRODUCTS:

Test Studio

24 125. Upon information and belief, and particularly by way of the detailed documentation
25 instructing users on how to use the Infringing Products in an infringing manner (*see supra*, ¶¶ 120,
26 124), Defendants have encouraged this infringement with knowledge of the '491 Patent and with a
27 specific intent to cause their customers and distributors to infringe.

126. Defendants' acts at least as of the filing of this Complaint thus constitute active inducement of patent infringement in violation of 35 U.S.C. § 271(b).

127. Defendants will, on information and belief, continue to induce infringement of the '491 Patent unless enjoined.

128. Defendants' direct infringement, contributory infringement, and inducement of infringement have irreparably harmed SRI.

129. Defendants will, on information and belief, continue to irreparably harm SRI unless enjoined.

130. Pursuant to 35 U.S.C. § 284, SRI is entitled to damages adequate to compensate for the infringement but in no event less than a reasonable royalty.

131. Defendants' infringement has been and is willful and, pursuant to 35 U.S.C. § 284, SRI is entitled to treble damages. Defendants' willful infringement is based at least on Defendants' knowledge of SRI, its products, and its patents since at least as early as the filing of this Complaint. Defendants have either willfully and wantonly infringed the '491 Patent or have recklessly avoided knowledge of their own infringement, even when faced with knowledge of SRI's own products and patents.

132. This case is "exceptional" within the meaning of 35 U.S.C. § 285, and SRI is entitled to an award of attorneys' fees.

COUNT VII – INFRINGEMENT OF THE '286 PATENT

133. SRI re-alleges and incorporates the allegations of the preceding paragraphs of this Complaint as if fully set forth herein.

134. SRI is the assignee and owner of all right, title, and interest in and to the '286 Patent, which was issued on November 26, 2019. A true and correct copy of the '286 Patent is attached hereto as Exhibit G.

135. The '286 Patent addresses an invention for testing websites. The disclosed innovation tests many facets of the website's experience and operation, including by providing novel approaches to creating, storing, and executing test scripts using website elements as opposed to the previously disclosed use of recording test scripts based upon user actions only.

136. SRI has the exclusive right to make, use, sell, and offer to sell any product embodying the '286 Patent throughout the United States, and to import any product embodying the '286 Patent into the United States.

137. SRI has commercially exploited the '286 Patent by making, marketing, selling, and using products covered by the '286 Patent, including its popular eValid™ software products. SRI continues to commercially exploit the '491 Patent through the present, at least by continuing to provide maintenance and support to users of its popular eValid™ software products.

138. Defendants have had knowledge of the '286 Patent, SRI, and SRI's products embodying the inventions claimed in the Patents-in-Suit since at least as early as the filing of this Complaint.

139. At all relevant times, SRI provided public notice of the '286 Patent by properly marking its products and its website pursuant to 35 U.S.C. § 287(a).

140. Defendants have been, and are currently, directly infringing at least claim 1 of the '286 Patent in violation of 35 U.S.C. § 271(a), literally or under the doctrine of equivalents, by making, using, selling, offering for sale, and/or importing into the United States Defendants' Infringing Products, which, as set forth in documentation available on Defendants' website, comprise the computing device disclosed in the '286 Patent—both as maintained in Defendants' files and as made accessible to its users to whom Defendants offer and sell the Infringing Products—including at least a memory; web browser program code stored in the memory; and a processor configured to perform the web browser program code, wherein the web browser program code, when performed, provides a web browser operating on the computing device; the Test Studio allows a user to browse the web via common web browsing activities, including navigating to a website and firing events such as clicking on a button (<https://docs.telerik.com/teststudio/features/recorder/overview>; <https://docs.telerik.com/teststudio/general-information/test-recording/overview>; <https://docs.telerik.com/teststudio/getting-started/first-project#test-recording>); wherein the web browser program code provides the web browser with Document Object Model (DOM) access capabilities; the Test Studio locates page elements based on their DOM indexes (<https://docs.telerik.com/teststudio/features/recorder/dom-explorer>;

<https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements>; <https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements>; <https://docs.telerik.com/teststudio/general-information/configure-your-browser/edge>); wherein the web browser program code, executable by the computing device, includes at least: computer program code for testing and analysis of a web page as rendered by the web browser (for example, the Test Studio allows for the creation of test scripts for testing and analysis of web pages by recording a user's interactions with the web page in question and allowing the user to play back those test scripts (<https://docs.telerik.com/teststudio/getting-started/first-project#test-recording>; <https://docs.telerik.com/teststudio/general-information/test-execution/quick-execution>); computer program code for accessing an attribute or property value of an element of a DOM of the web page, wherein the computer program code for accessing the attribute or property value of the element of the DOM of the web page accesses the DOM of the web page using a browser programming interface that enables the web browser program code to have access to the DOM; the Test Studio locates page elements based on their DOM indexes (<https://docs.telerik.com/teststudio/features/recorder/dom-explorer>; <https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements>; <https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements>; <https://docs.telerik.com/teststudio/general-information/configure-your-browser/edge>); the browser programming interface is supported by an API underlying the web browser program code for providing a plurality of library function calls or methods that are accessible by the web browser program code; the Test Studio includes a browser programming interface capable of accessing an underlying API (<https://docs.telerik.com/teststudio/features/execute-apitest/add-api-test-as-step>; <https://www.telerik.com/blogs/power-up-your-ui-tests-with-api-test-as-step-in-telerik-test-studio>; <https://docs.telerik.com/teststudio/general-information/configure-your-browser/edge>); and wherein the computer program code for accessing the attribute or property value of the element of the DOM of the web page accesses the attribute or property value of the element of the DOM of the web page

1 for purposes of the testing and analysis of the web page rendered in the web browser; the Test Studio
 2 interrogates the DOM to identify and extract relevant information regarding at least the page
 3 elements germane to the script, including each such element's index and value, and stores those
 4 details in the test script ([https://docs.telerik.com/teststudio/general-information/test-results/analyze-
 5 test-list-results](https://docs.telerik.com/teststudio/general-information/test-results/analyze-test-list-results)); the Test Studio uses explicit wait commands that "wait for the comparison to be true
 6 before proceeding"—in order to perform such a validation, the Test Studio must necessarily store
 7 facts about the webpage being rendered, i.e., the expected condition to be checked for during
 8 validation (<https://docs.telerik.com/teststudio/features/recorder/verifications/wait>;
 9 <https://www.telerik.com/blogs/explicit-waits-with-ajax-combobox-example>); wherein the web
 10 browser program code supports at least one command, provided to the web browser via the browser
 11 programming interface, to facilitate synchronized testing and analysis of asynchronous processes of
 12 the web page rendered by the web browser using the underlying API; the Test Studio allows for the
 13 testing of content dynamically generated by AJAX programming using, for example, its various wait
 14 commands to synchronize playback and allow for testing of content dynamically generated by AJAX
 15 programming (<https://docs.telerik.com/teststudio/features/recorder/verifications/wait>;
 16 <https://www.telerik.com/blogs/explicit-waits-with-ajax-combobox-example>); and wherein the at least
 17 one command includes a DOM index value, a DOM property name and a DOM property value, and
 18 causes examination of a name and a value of a property found in the DOM of the web page at the
 19 DOM index value to determine whether the name and the value match the DOM property name and
 20 the DOM property value, respectively; to generate and subsequently perform validation tests, the Test
 21 Studio interrogates the DOM to identify and extract relevant information regarding at least the page
 22 elements germane to the script, including each such element's index and value, and stores those
 23 details in the test script ([https://docs.telerik.com/teststudio/general-information/test-results/analyze-
 24 test-list-results](https://docs.telerik.com/teststudio/general-information/test-results/analyze-test-list-results)); the Test Studio uses explicit wait commands that "wait for the comparison to be true
 25 before proceeding"—in order to perform such a validation, the Test Studio must necessarily store
 26 facts about the webpage being rendered, i.e., the expected condition to be checked for during
 27 validation (<https://docs.telerik.com/teststudio/features/recorder/verifications/wait>;
 28 <https://www.telerik.com/blogs/explicit-waits-with-ajax-combobox-example>); the Test Studio locates

1 page elements based on their DOM indexes, using DOM access to test dynamic web applications,
2 including those written in AJAX (<https://docs.telerik.com/teststudio/features/recorder/dom-explorer>;
3 [https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements)
4 [wtc/element-identification-wtc/finding-page-elements](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements); [https://docs.telerik.com/teststudio/testing-](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements)
5 [framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements)
6 [elements](https://docs.telerik.com/teststudio/testing-framework/write-tests-in-code/intermediate-topics-wtc/element-identification-wtc/finding-page-elements); <https://docs.telerik.com/teststudio/general-information/configure-your-browser/edge>), as
7 disclosed in the '286 Patent.

8 141. Defendants will, on information and belief, continue to directly infringe the '286
9 Patent unless enjoined.

10 142. To the extent Defendants' Infringing Products, without more, do not directly infringe
11 at least claim 1 of the '286 Patent, at least as of the filing of this Complaint, Defendants contribute to
12 infringement of the same under 35 U.S.C. § 271(c) inasmuch as the Infringing Products offered for
13 sale and sold by Defendants are each a component of a patented machine or an apparatus used in
14 practicing a patented process, constituting a material part of SRI's invention, knowing the same to be
15 especially made or especially adapted for use in infringement of the '286 Patent. For example, as set
16 forth above, the Test Studio, when used in its normal and intended usage (pursuant to the instructions
17 set forth on Defendants' website), infringes claim 1 of the '286 Patent. *See supra*, ¶ 140.

18 143. Defendants will, on information and belief, continue to contribute to infringement of
19 the '286 Patent unless enjoined.

20 144. Defendants actively encourage their customer to use Defendants' Infringing Products
21 in an infringing manner. For example, Defendants' website is replete with written directions,
22 screenshots, and videos instructing users on how to use the Infringing Products in an infringing
23 manner. For example, as set forth above, Defendants' website regarding the Test Studio specifically
24 instructs users of the Infringing Products how to infringe claim 1 of the '286 patent. *See supra*,
25 ¶ 140. Defendants' website also touts the identities of customers who use the Infringing Products,
26 each of whom is a direct infringer inasmuch as they use the Infringing Products in the infringing
27 manner as instructed by Defendants:
28

Mark Judson
SOFTWARE DEVELOPER,
EBSCO INDUSTRIES INC.

“ After using the trial for a week we increased our automated testing from 5% to 30%. Test Studio was so easy to incorporate into our existing testing environment! Tests are stable and easy to run and understand.

RELATED PRODUCTS: [Functional Testing - SILVERLIGHT](#) [Test Studio](#)

Ronak Samantray
SOFTWARE DEVELOPER,
NOWFLOATS TECHNOLOGIES

“ Love the controls! They make my life so simple. The best part is the product is indeed a WYSIWIG :) - it delivers what it promises.

RELATED PRODUCTS: [Telerik UI for Windows 8](#) [Test Studio](#) [TeamPulse](#)

Amit Deshpande
TECHNICAL LEADER,
ITRA

“ Recently I downloaded Test Studio. Coming from a development background, I know how important is it to completely test any application before delivering it to the client. I was really amazed at the robustness of Test Studio. I can perform Functional, Performance, Load, etc. in one product. Earlier we were focusing on many products out there for each type of testing, however now its just one package .. Just Love it ..

RELATED PRODUCTS: [Test Studio](#)

145. Upon information and belief, and particularly by way of the detailed documentation instructing users on how to use the Infringing Products in an infringing manner (*see supra*, ¶¶ 140, 144), Defendants have encouraged this infringement with knowledge of the ‘286 Patent and with a specific intent to cause their customers and distributors to infringe.

146. Defendants’ acts at least as of the filing of this Complaint thus constitute active inducement of patent infringement in violation of 35 U.S.C. § 271(b).

147. Defendants will, on information and belief, continue to induce infringement of the ‘286 Patent unless enjoined.

148. Defendants’ direct infringement, contributory infringement, and inducement of infringement have irreparably harmed SRI.

149. Defendants will, on information and belief, continue to irreparably harm SRI unless enjoined.

150. Pursuant to 35 U.S.C. § 284, SRI is entitled to damages adequate to compensate for the infringement but in no event less than a reasonable royalty.

151. Defendants’ infringement has been and is willful and, pursuant to 35 U.S.C. § 284, SRI is entitled to treble damages. Defendants’ willful infringement is based at least on Defendants’ knowledge of SRI, its products, and its patents since at least as early as the filing of this Complaint.

1 Defendants have either willfully and wantonly infringed the '286 Patent or have recklessly avoided
2 knowledge of their own infringement, even when faced with knowledge of SRI's own products and
3 patents.

4 152. This case is "exceptional" within the meaning of 35 U.S.C. § 285, and SRI is entitled
5 to an award of attorneys' fees.

6 **DEMAND FOR JURY TRIAL**

7 SRI hereby demands a trial by jury of all issues so triable under Federal Rule of Civil
8 Procedure 38(b).

9 **PRAYER FOR RELIEF**

10 WHEREFORE, SRI respectfully requests that this Court:

- 11 A. Find that United States Patent No. 7,757,175 is valid and enforceable against
12 Defendants;
- 13 B. Find that Defendants have infringed and are infringing United States Patent No.
14 7,757,175;
- 15 C. Permanently enjoin Defendants, their officers, agents, servants, employees, and those
16 persons acting in active concert or in participation therewith from infringing United
17 States Patent No. 7,757,175;
- 18 D. Award SRI damages sufficient to compensate it for Defendants' past and future
19 infringement of United States Patent No. 7,757,175, together with costs and
20 prejudgment interest, pursuant to 35 U.S.C. § 284;
- 21 E. Find that United States Patent No. 8,327,271 is valid and enforceable against
22 Defendants;
- 23 F. Find that Defendants have infringed and are infringing United States Patent No.
24 8,327,271;
- 25 G. Permanently enjoin Defendants, their officers, agents, servants, employees, and those
26 persons acting in active concert or in participation therewith from infringing United
27 States Patent No. 8,327,271;
- 28

- 1 H. Award SRI damages sufficient to compensate it for Defendants' past and future
2 infringement of United States Patent No. 8,327,271, together with costs and
3 prejudgment interest, pursuant to 35 U.S.C. § 284;
- 4 I. Find that United States Patent No. 8,392,890 is valid and enforceable against
5 Defendants;
- 6 J. Find that Defendants have infringed and are infringing United States Patent No.
7 8,392,890;
- 8 K. Permanently enjoin Defendants, their officers, agents, servants, employees, and those
9 persons acting in active concert or in participation therewith from infringing United
10 States Patent No. 8,392,890;
- 11 L. Award SRI damages sufficient to compensate it for Defendants' past and future
12 infringement of United States Patent No. 8,392,890, together with costs and
13 prejudgment interest, pursuant to 35 U.S.C. § 284;
- 14 M. Find that United States Patent No. 8,495,585 is valid and enforceable against
15 Defendants;
- 16 N. Find that Defendants have infringed and are infringing United States Patent No.
17 8,495,585;
- 18 O. Permanently enjoin Defendants, their officers, agents, servants, employees, and those
19 persons acting in active concert or in participation therewith from infringing United
20 States Patent No. 8,495,585;
- 21 P. Award SRI damages sufficient to compensate it for Defendants' past and future
22 infringement of United States Patent No. 8,495,585, together with costs and
23 prejudgment interest, pursuant to 35 U.S.C. § 284;
- 24 Q. Find that United States Patent No. 8,650,493 is valid and enforceable against
25 Defendants;
- 26 R. Find that Defendants have infringed and are infringing United States Patent No.
27 8,650,493;
- 28

- 1 S. Permanently enjoin Defendants, their officers, agents, servants, employees, and those
2 persons acting in active concert or in participation therewith from infringing United
3 States Patent No. 8,650,493;
- 4 T. Award SRI damages sufficient to compensate it for Defendants' past and future
5 infringement of United States Patent No. 8,650,493, together with costs and
6 prejudgment interest, pursuant to 35 U.S.C. § 284;
- 7 U. Find that United States Patent No. 8,984,491 is valid and enforceable against
8 Defendants;
- 9 V. Find that Defendants have infringed and are infringing United States Patent No.
10 8,984,491;
- 11 W. Permanently enjoin Defendants, their officers, agents, servants, employees, and those
12 persons acting in active concert or in participation therewith from infringing United
13 States Patent No. 8,984,491;
- 14 X. Award SRI damages sufficient to compensate it for Defendants' past and future
15 infringement of United States Patent No. 8,984,491, together with costs and
16 prejudgment interest, pursuant to 35 U.S.C. § 284;
- 17 Y. Find that United States Patent No. 10,489,286 is valid and enforceable against
18 Defendants;
- 19 Z. Find that Defendants have infringed and are infringing United States Patent No.
20 10,489,286;
- 21 AA. Permanently enjoin Defendants, their officers, agents, servants, employees, and those
22 persons acting in active concert or in participation therewith from infringing United
23 States Patent No. 10,489,286;
- 24 BB. Award SRI damages sufficient to compensate it for Defendants' past and future
25 infringement of United States Patent No. 10,489,286, together with costs and
26 prejudgment interest, pursuant to 35 U.S.C. § 284;
- 27 CC. Order an accounting of damages from Defendants' infringement;
- 28

- 1 DD. Award SRI enhanced damages, up to and including trebling SRI's damages, pursuant
2 to 35 U.S.C. § 284, for Defendants' willful infringement of the Patents-in-Suit;
3 EE. Award SRI its reasonable attorney fees and costs of suit pursuant to 35 U.S.C. § 285
4 due to the exceptional nature of this case, or as otherwise permitted by law;
5 FF. Award SRI post-judgment interest pursuant to 28 U.S.C. § 1961; and
6 GG. Award SRI such other or additional relief as the Court deems just and proper.
7

8 Date: March 16, 2020

Respectfully submitted,

9 SINGER CASHMAN LLP

10
11 By: 

Benjamin L. Singer

Evan Budaj

Attorneys for Plaintiff Software Research, Inc.